



Université Paris Cité

UNIVERSITÉ PARIS CITÉ

UE PROJET DE PROGRAMMATION
PROJET L2I1 - TAMAGOTCHI

Conception détaillée L2I1 - Tamagotchi

Auteurs :

Marwan DENAGNON
Lina BOUGUETTAYA
Yasmine DEHOUCHE
Abhijeet SINGH

Encadrant :

Camille KURTZ

Responsable de l'UE :

David JANISZEK

9 mars 2025

Sommaire

1	Introduction	3
1.1	Objectifs et méthodes	3
1.2	Documents de référence	3
2	Guide de lecture	3
2.1	Maîtrise d'œuvre	3
2.1.1	Responsable	3
2.1.2	Personnel technique	4
2.2	Maîtrise d'ouvrage	4
3	Architecture logiciel	4
4	Technologies et outils	4
4.1	Front end	4
4.2	Back end	4
4.3	Gestion de version et collaboration	5
5	Description des modules	5
5.1	Page de choix de connexion ou d'inscription	5
5.2	Page de connexion	5
5.3	Page d'inscription	5
5.4	Page de chargement	5
5.5	Page d'accueil	5
5.6	Page d'information de l'animal et de l'utilisateur	6
5.7	Page de paramètres	6
5.8	Page de transfert NFC	6
5.9	Page de gestion des données	6
5.10	Page d'états de l'animal	6
5.10.1	État de faim	6
5.10.2	État de saleté	6
5.10.3	État d'ennui	6
5.10.4	État de sommeil	6
5.10.5	État de maladie	7
5.10.6	État de soif	7
5.10.7	État de négligence	7
5.10.8	État de mort	7
6	Arborescence de l'application	8
7	Diagramme de cas d'utilisation	9
7.1	Diagramme de connexion	10
8	Diagramme de classes	11
9	Conception de la base de données (Modèle ERD)	12

10 Diagramme : Cycle de vie	13
11 Organisation	14
12 Planification	14
12.1 Planning des projets	14
12.2 Diagramme de Gantt de la planification	15
13 Glossaire	15

1 Introduction

Ce document constitue **la conception détaillée** de notre projet de programmation L2I1 : Simulation de vie d'un petit animal. Cette **conception détaillée** a pour objectif de définir, formaliser et expliquer les aspects techniques tout en permettant de vérifier et d'indiquer l'architecture de l'application. On y trouve la définition des différentes fonctions, classes, librairie et autres structures dont on a besoin pour développer cette application. Ce document contiendra plusieurs parties, nous désignons d'abord les objectifs de base à la compréhension de notre projet, ensuite nous enchaînons avec l'architecture de logicielle, ainsi que les technologies et outils qui seront utilisées lors de l'étape de développement, nous y trouvons également la partie planification et l'organisation des tâches qui est primordiale pour la fluidité et structuration de cette phase de développement. Elle contient également la schématisation des structures prête à ce que cela soit facilement traduit en code source. Ce document servira de référence tout au long du développement du projet afin d'assurer une conception structurée et rigoureuse.

1.1 Objectifs et méthodes

Notre objectif primaire est de créer une application android fonctionnelle pour prendre soin d'un animal de compagnie virtuel, voire deux. Ce dernier sera nourri, lavé, diverti et soigné pour répondre à ses besoins afin de le garder en vie. Si le maître de l'animal n'est pas disponible pour cela, nous prévenons un système NFC pour transférer la créature à quelqu'un qui pourra s'en occuper entre temps. Le contenu de ce document nous permettra de mettre un doigt sur les détails de conception techniques.

1.2 Documents de référence

Les documents de référence de ce projet sont :

- Le descriptif du projet.
- <https://www.ens.math-info.univ-paris5.fr/projets-informatiques/doku.php?id=projets:licence2:2024-2025>
- Le cahier des charges.
- Le cahier de recette
- La maquette de l'application
- Les comptes rendus de chaque réunion.

2 Guide de lecture

2.1 Maîtrise d'œuvre

2.1.1 Responsable

Nom de l'encadrant : Kurtz Camille

Le maître d'œuvre supervise et apporte un soutien aux propositions de l'équipe concernant le développement de l'application android. Il s'occupe de la validation des documents principaux.

2.1.2 Personnel technique

Le personnel du projet est constitué des quatres étudiants choisie :

- Lina BOUGUETTAYA
- Yasmine DEHOUCHE
- Marwan DENAGNON
- Abhijeet SINGH

2.2 Maîtrise d'ouvrage

Le maître d'ouvrage est le client de notre projet. Ce dernier est représenté par l'encadrant du groupe M. KURTZ.

3 Architecture logiciel

L'architecture est basée sur MVVM (Model-View-ViewModel), un modèle populaire dans les applications Android modernes, qui permet une séparation claire des responsabilités tout en facilitant la gestion de l'état de l'application.

- Model : Représente les données de l'application et la logique métier.
- View : Gère l'interface utilisateur (UI).
- ViewModel : Gère les interactions entre la View et le Model, et contient la logique de présentation.

4 Technologies et outils

4.1 Front end

Utilise des outils et technologies qui permettent la modélisation de l'application :

- Figma : outil de conception visuelle utilisé pour la réalisation de maquettes interactives de l'interface utilisateur.
- Canva : plateforme graphique utilisée pour la création d'assets graphiques destinés à l'application.
- XML (Extensible Markup Language) : langage de balisage utilisé pour structurer, stocker et échanger les données de manière hiérarchique.

4.2 Back end

Repose sur un ensemble d'outils afin d'assurer la gestion et l'interaction avec l'interface utilisateur :

- Android Studio : environnement de développement intégré officiel pour Android ayan des outils avancés pour l'édition du code, le débogage et l'émulation de l'application.
- Java : langage de programmation utilisé pour implémenter les fonctionnalités de l'application.

- Android SDK : Ensemble d'outils fournis par google pour le développement d'applications Android.
- JSON (JavaScript Object Notation) : représentation structurée des données sous forme de paires clé-valeur, utilisée pour échanger des informations entre un client et un serveur.

4.3 Gestion de version et collaboration

Se base sur des outils permettant une coordination fluide entre les membres de l'équipe afin de garantir la qualité du code tout au long du développement :

- Apache Subversion (SVN) : système permettant le suivi des modifications du code source et la collaboration entre les membres de l'équipe.
- GitHub : plateforme de gestion et de collaboration permettant l'hébergement du code source

5 Description des modules

5.1 Page de choix de connexion ou d'inscription

- Bouton **“Se créer un compte”**
- Bouton **“J’ai déjà un compte”**

5.2 Page de connexion

- Champs **Identifiant** et **Mot de Passe**
- Bouton **“Connexion”**

5.3 Page d'inscription

- Champs **Nom** et **Prénom**
- Champ **Date de naissance**
- Champ **Genre de l’animal**
- Bouton **“S’inscrire”**

5.4 Page de chargement

- Barre de chargement

5.5 Page d'accueil

- Barre d'état de l'animal
- Bulle de discussion **“Salut”**
- Barre de tâches **“Soigner, Manger...”**

5.6 Page d'information de l'animal et de l'utilisateur

- Fiche d'information concernant l'animal et son maître
- Champ du **Nom de l'animal** (possibilité de le changer)
- Bouton de **Paramètres**

5.7 Page de paramètres

- Switch de synchronisation NFC
- Switch de désactivation des effets sonores
- Switch de désactivation des animations
- Switch “**Ajout d'un widget à l'écran d'accueil**”
- Bouton de **Déconnexion**

5.8 Page de transfert NFC

- Bouton de confirmation “**Transférer vos données**”

5.9 Page de gestion des données

- Création ou téléchargement d'un **Backup local**
- Exportation des données
- Bouton **Suppression des données**

5.10 Page d'états de l'animal

5.10.1 État de faim

- Bulle de discussion “**J'ai faim**”
- Diminution de la barre de **Nourriture**

5.10.2 État de saleté

- Bulle de discussion “**Je veux un bain**”
- Diminution de la barre de **Propreté**
- Affichage d'un animal sale

5.10.3 État d'ennui

- Bulle de discussion “**Je m'ennuie ! joue avec moi**”
- Diminution de la barre de **Loisir**
- Affichage d'un animal qui s'ennuie

5.10.4 État de sommeil

- Bulle de discussion “**Je veux dormir**”
- Diminution de la barre de **Sommeil**
- Affichage d'un animal fatigué

5.10.5 État de maladie

- Bulle de discussion “**J’ai bobo**”
- Diminution de la barre de **Santé**
- Affichage d’un animal malade

5.10.6 État de soif

- Bulle de discussion “**J’ai soif!**”
- Diminution de la barre de **Soif**
- Affichage d’un animal qui a soif

5.10.7 État de négligence

- Bulle de discussion “**Je me sens délaissé! prends soin de moi**”
- Diminution de toutes les barres d’états
- Affichage d’un animal en mauvaise mine

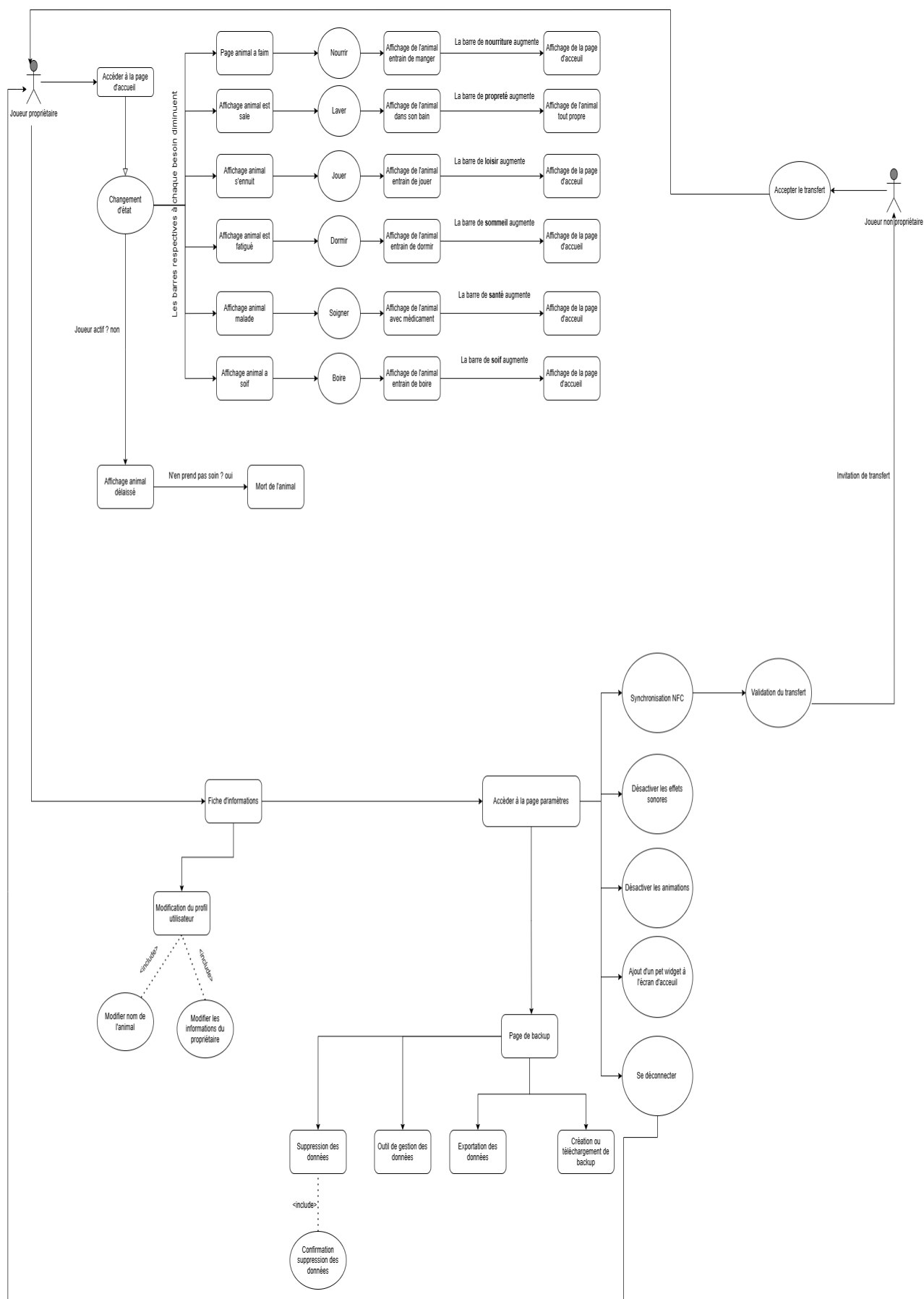
5.10.8 État de mort

- Toutes les barres sont à zéro
- Affichage d’un animal mort

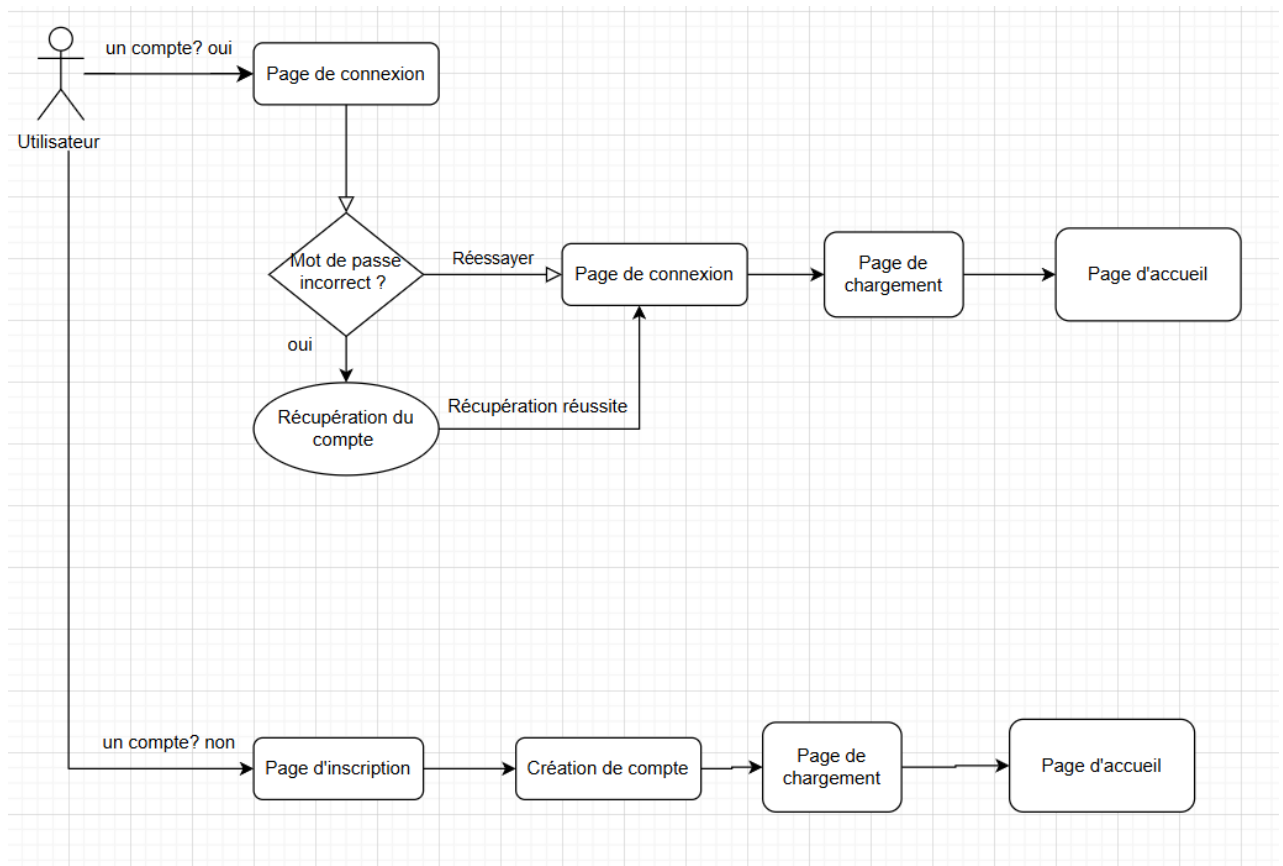
6 Arborescence de l'application



7 Diagramme de cas d'utilisation

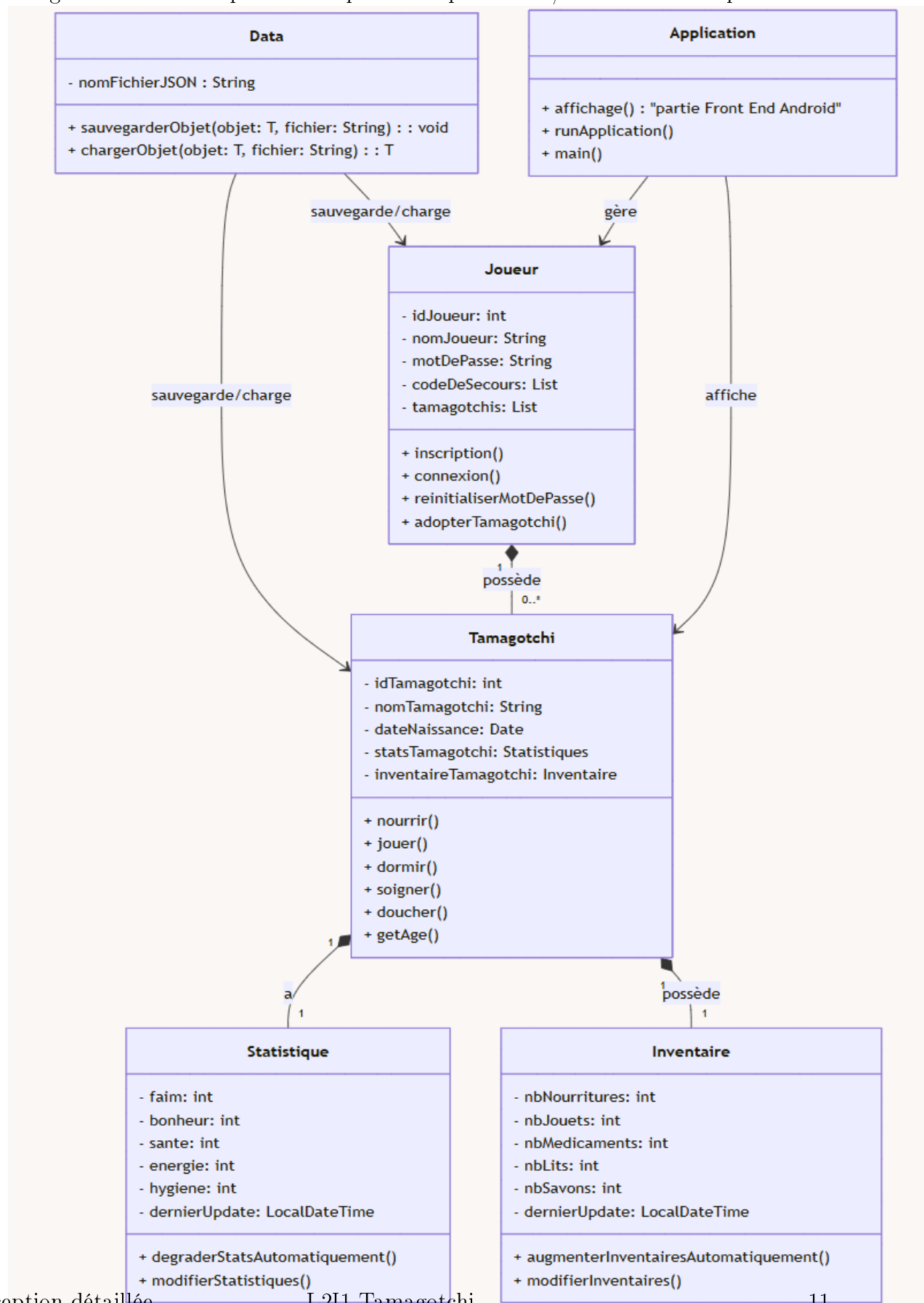


7.1 Diagramme de connexion

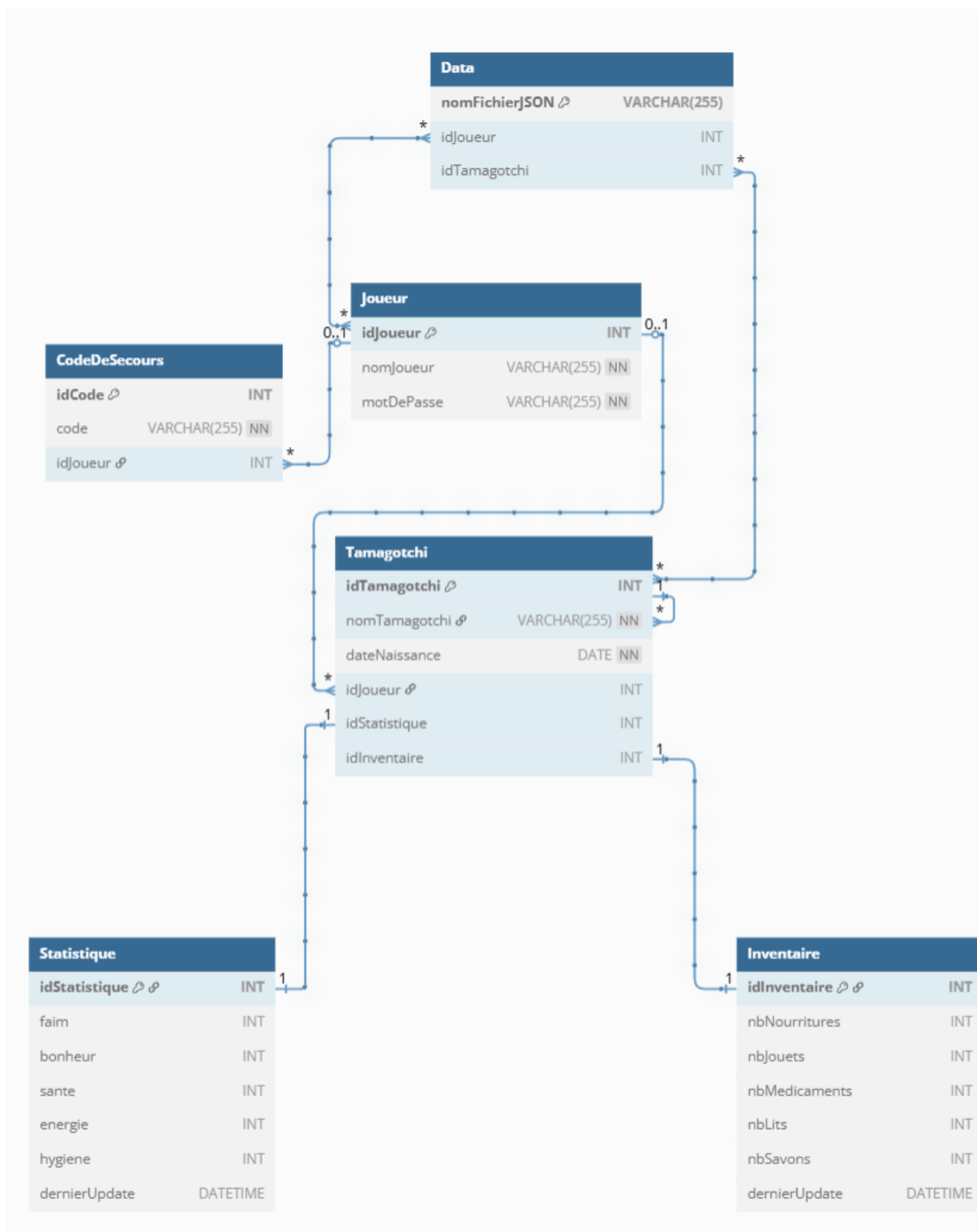


8 Diagramme de classes

Les getters et setters qui ne font qu'une simple lecture/écriture ne sont pas inclus.

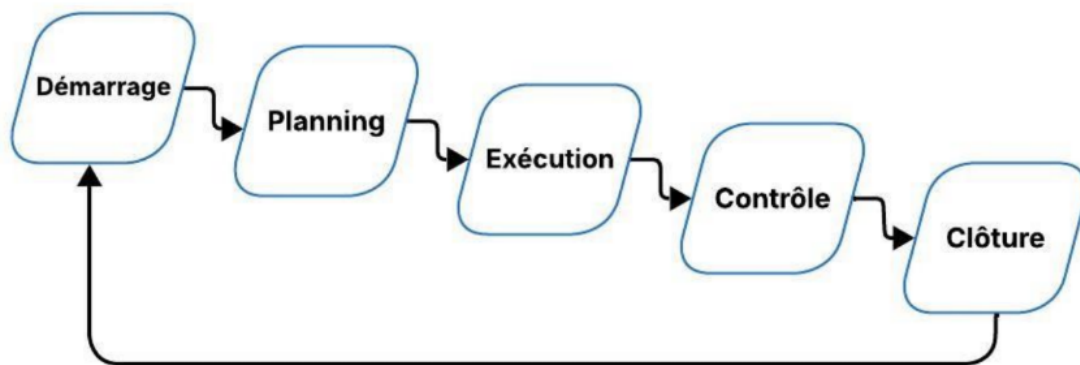


9 Conception de la base de données (Modèle ERD)



10 Diagramme : Cycle de vie

Le cycle de vie d'un projet décrit l'ensemble des processus nécessaires à sa réalisation. Il se compose de différentes étapes, illustrées par le schéma général ci-dessous :



Démarrage (initialisation) : identification des besoins et validation du lancement du projet.

Planning : précision des objectifs et organisation des tâches, des ressources et des délais nécessaires.

Exécution : réalisation du travail défini précédemment pour atteindre les objectifs.

Contrôle (surveillance) : vérification de l'avancement et ajustement des écarts si nécessaire.

Clôture : finalisation du projet, évaluation des résultats et archivage des documents.

11 Organisation

Le tableau ci-dessous représente l'organisation du projet sous forme de décomposition des tâches :

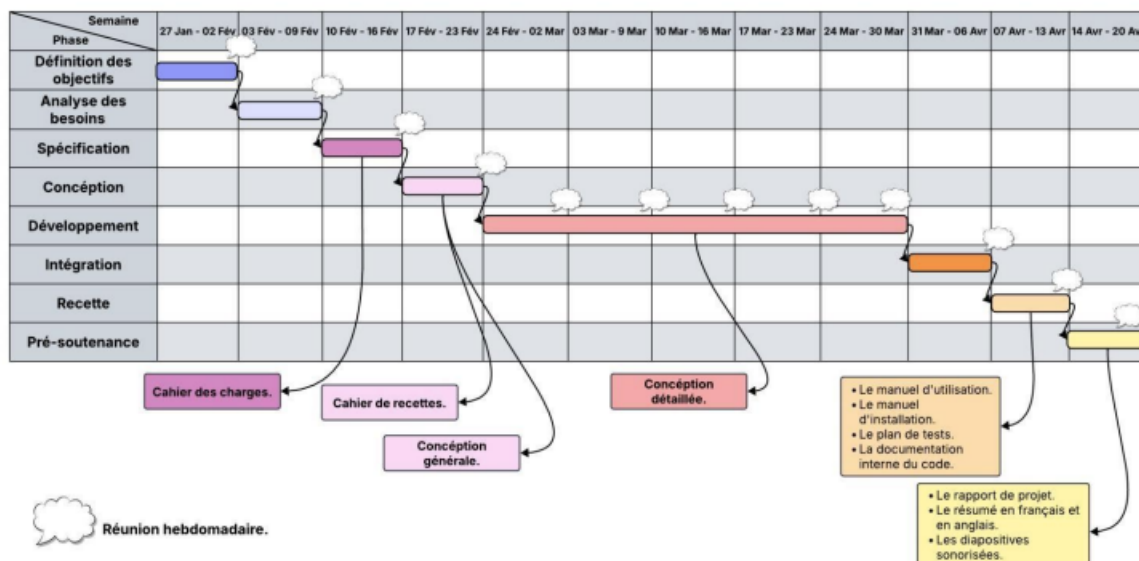
Id	Tâche
1	Développement Frontend des pages principales de l'application : - Création de maquettes pour l'interface utilisateur (UI) ;
2	Développement Frontend : - Implémentation de l'UI (boutons, états, actions) ;
3	Développement Backend : - Implémentation de la création et de la gestion de l'animal (interactions et états) ;
4	Développement Backend : - Création et gestion de la base de données pour stockage ;
5	Développement Frontend : - Implémentation des animations de l'animal ;
6	Développement Backend et Frontend : - Intégration des fonctionnalités en reliant le Backend aux éléments du Frontend ;
7	Développement Backend et Frontend : - Tests unitaires des fonctionnalités et débogage si nécessaire ;
8	Développement Backend et Frontend : - Tests finaux après débogage et optimisation ;
9	Développement Backend et Frontend : - Préparation de l'application pour déploiement et lancement ;

TABLE 1 – Organisation des tâches du projet

12 Planification

12.1 Planning des projets

Le projet s'étend sur une période de 12 semaines, ponctuée de réunions hebdomadaires. Il se conclut par des soutenances et la remise d'un rapport. Les quatre premières semaines sont consacrées à la documentation, comprenant la rédaction du cahier des charges, du cahier de recettes et du plan de tests. Les six semaines suivantes sont dédiées au développement. Enfin, les deux dernières semaines seront réservées à la rédaction du rapport et à la préparation de la soutenance. Le diagramme ci-dessous montre la répartition des tâches sur la durée du projet :



12.2 Diagramme de Gantt de la planification

Semaine Tâche	Semaine 5	Semaine 6	Semaine 7	Semaine 8	Semaine 9	Semaine 10	Semaine 11
Tâche 1							
Tâche 2							
Tâche 3							
Tâche 4							
Tâche 5							
Tâche 6							
Tâche 7							
Tâche 8							
Tâche 9							

13 Glossaire

- Arborescence : Structure hiérarchique représentant l'organisation ou la disposition des éléments dans un système.
- Diagramme : Représentation visuelle utilisée pour illustrer des concepts, des processus ou des relations entre des éléments.

- Diagramme de Gantt : Outil de gestion de projet utilisé pour planifier et suivre les tâches d'un projet dans le temps
- Front-end : Partie visible d'un site ou d'une application avec laquelle l'utilisateur interagit ici tous ce qui est visible sur l'application.
- Back-end : Partie cachée qui gère la logique, les bases de données et le serveur, ici le fichier JSON
- Joueur non propriétaire : C'est l'utilisateur de l'appareil auquel le transfert sera fait pour s'occuper de l'animal lors de l'absence du maitre